



Hartree Centre

Science & Technology Facilities Council

<http://tinyurl.com/CP2KSchool2018>

[#CP2KSummerSchool](#)

BUILDING CP2K

CP2K Summer School, 19-22 June 2018

Iain Bethune

iain.bethune@stfc.ac.uk

@iainbethune





Overview

- CP2K Toolchain
- Prerequisites
 - Environment
 - Libraries
- Optional Libraries
 - Functionality
 - Performance
- Arch files and compilation
- Testing CP2K





CP2K Toolchain

- The one-hit wonder build script!
- Attempts a basic build of CP2K – works well on standard Linux systems

```
./install_cp2k_toolchain.sh
```

or

```
./install_cp2k_toolchain.sh --install-all  
then cross your fingers!
```

- If you have an ‘unusual’ system e.g. a cluster or HPC...

```
./install_cp2k_toolchain.sh --help
```





Prerequisites - Environment

- POSIX-compliant OS
 - Linux, UNIX (e.g. AIX) ...
 - Cygwin, Mac OS X also possible
- Build tools
 - GNU Make, Python 2.x (or later)
- Compilers (see https://www.cp2k.org/dev:compiler_support)
 - GNU gcc / gfortran 4.6 (or later)
 - Intel ifort 15.x (or later, with care!)
 - IBM XLF 14.1
 - Cray 8.7.0





Prerequisites - Libraries

- BLAS & LAPACK (required)
 - Vendor-tuned libraries preferred (MKL, ACML, ESSL)
 - Free auto-tuned libraries (GotoBLAS, ATLAS)
 - Reference BLAS + LAPACK from Netlib (last resort, very slow!)
- MPI & ScaLAPACK (required for MPI parallel build)
 - Usually provided by your cluster / HPC
 - Require MPI 2.x (3.x optional)
 - OpenMPI, MPICH, Intel MPI, Cray MPT all tested
 - ScaLAPACK provided by vendor maths libraries...
 - ... or download from Netlib
 - `-D__parallel -D__SCALAPACK`





Prerequisites - Libraries

- FFTW3 (Recommended)
 - CP2K has an inbuilt FFT implementation
 - FFTW3 will give much better performance
 - + freely available
 - + easy to compile / install
 - Enable using `-D__FFTW3`





Optional Libraries

- Libxc
 - CP2K has various common XC functionals e.g. PBE, LDA, BLYP...
 - Many more available via libxc
 - Version 4.0.3 or later
 - `-D__LIBXC`
- Libint
 - Required for all Hartree-Fock Exchange calculations
 - Version 1.1.4 only
 - `-D__LIBINT`





Optional Libraries

- ELPA
 - Optimised diagonalisation routines
 - Build process optimises for specific architecture
 - Specify ELPA version : `-D__ELPA=YYYYMM`
- Auto-tuned performance libraries (lib[x]smm, libgrid)
- All other libraries / options / flags
 - See <http://www.cp2k.org/howto:compile>
 - and `cp2k/INSTALL`





Arch files and compilation

- Compiler and architecture-specific options are given in an 'arch file'
 - Examples in `cp2k/arch`
 - e.g. `Linux_x86-64-gfortran.popt`
 - Copy/customise for your environment

- To build CP2K
 - in the `cp2k/makefiles` directory:

corresponding to arch file

- `make -j 4 ARCH=Linux-x86-64-gfortran VERSION=popt`

parallel build

Errors? Ask us!





Arch files and compilation

- CP2K binary should be built in
`cp2k/exe/<ARCH>/cp2k.<VERSION>`
- Very quick test:
`cp2k.sopt --version`
 - MPI binaries (`popt`) should be run with `mpirun/aprun/mpiexec`
 - Maybe within a batch script?
- Quick test
 - in the `cp2k/tests/QS` directory:
`../../../../exe/ARCH/cp2k.sopt C.inp`





Testing CP2K

- CP2K comes with a suite of >3000 test input files
- Good for checking you have correctly compiled CP2K
 - Tests that all enabled features of CP2K run
 - Most tests compare against a reference result
- To execute regression tests:
 - Instructions in `cp2k/tools/regtesting`
 - Also online: <http://cp2k.org/dev:regtesting>





Testing CP2K

- `do_regtest` script
 - SVN update, builds CP2K (`--nosvn -nobuild` to skip)
 - Runs all tests (in parallel, if possible)
 - Takes ~10 mins – a few hours
 - Summary of results and details of any failing tests

```
----- Summary -----  
Number of COMPILE warns 0  
Number of FAILED tests 2  
Number of WRONG tests 51  
Number of CORRECT tests 2589  
Number of NEW tests 0  
Total number of tests 2642
```

Test failed to complete

Test completed, but does not match reference

Test completed for first time (and no reference result available)





Testing CP2K

- Automatic testing on 30+ different platforms
 - Test failures automatically reported to developers
- Results available online at <http://dashboard.cp2k.org>
- Check here when using an SVN trunk version





Hartree Centre

Science & Technology Facilities Council

Building CP2K

Try it...

Questions?

CP2K